AMPOLA: Criação de Cenários de Análise de *Malware* focados em *Red Team* via Orquestração de Ambientes

1

Abstract. *In this article, we introduce AMPOLA, a tool developed for attack analysis through the orchestration of components that simulate complex scenarios in controlled environments. We validated AMPOLA in realistic scenarios, including multi-component malware analysis, to demonstrate its ability to observe the operation of sophisticated threats.*

Resumo. *Este artigo apresenta o sistema AMPOLA, desenvolvido para análise de ataques por meio da orquestração de componentes que simulam cenários complexos em ambientes controlados. Validou-se AMPOLA em cenários realistas, incluindo análise de malware multi-componente, de forma a mostrar sua capacidade de observar o funcionamento de ameaças sofisticadas.*

1. Introdução

O cenário de ameaças cibernéticas está em constante evolução, com a proliferação de exemplares de malware cada vez mais sofisticados e evasivos, fazendo com que as organizações enfrentem desafios crescentes para detectar e responder a esses tipos de ataque, dada a complexidade e modularidade dos programas maliciosos atuais [Gatlan 2024]. Com isso, tem-se a demanda por soluções mais customizáveis para aumentar a eficácia da análise, considerando-se a orquestração de segurança, a simulação de ataque de *red team*, a caçada de ameaças (*threat hunting*) e análise automatizada. A orquestração de segurança visa automatizar e integrar tarefas de segurança, otimizando o fluxo de trabalho e aprimorando a eficiência das equipes de segurança. Os exercícios de *red team* simulam ataques reais em ambientes controlados, permitindo que as organizações identifiquem e corrijam vulnerabilidades antes que sejam exploradas por atacantes. A caçada de ameaças é um processo proativo de busca por indicadores de comprometimento (IOC) e outras evidências de atividade maliciosa em uma rede ou sistema. Ao combinar essas áreas, fomenta-se a criação de infraestruturas simuladas que sirvam para analisar e compreender ataques por *malware* multicomponente.

Neste artigo, propõe-se o sistema AMPOLA (Análise de Malware por Pilotagem via Orquestração Lógica de Ambientes), cujo objetivo é auxiliar nas tarefas de investigação de incidentes, reduzindo o tempo e o esforço necessários para identificação da causa raiz dos ataques e tomada de contra-medidas, bem como aprimorar a capacidade de analisar e responder a malware complexo. Para tanto, utiliza-se de técnicas de orquestração e automação de ambientes virtuais propícios a execução deste tipo de amostra simulando cenários de infecção realistas.As contribuições deste trabalho são: (i) introduzir o sistema "AMPOLA", detalhando sua arquitetura, componentes e decisões de projeto; (ii) validar o AMPOLA a partir de estudos de caso que permitam a observação de infecções via integração de múltiplos ambientes em cenários de infecção por *malware*; (iii) disponibilizar o AMPOLA para a comunidade usar e contribuir com sua evolução.

2. Trabalhos Relacionados

Ferramentas tradicionais para criação de cenários de *red team* focam em fornecer capacidades avançadas de pós-exploração, permitindo que os times realizem ataques simulados detalhados e complexos. No entanto, essas ferramentas geralmente requerem um alto nível de interação manual e experiência técnica.

[Ray et al. 2005] propõem um modelo automatizado para simulação de ataques cibernéticos visando auxiliar na identificação de vulnerabilidades em sistemas. Utilizando casos de uso baseados em UML, diagramas de sequência e de estado, e XML, o modelo facilita a automação de ataques e sua documentação detalhada.

[Lee et al. 2022] focam na automatização das tarefas repetitivas do processo de resposta a incidentes para fornecer resposta rápida com influência humana mínima, expandindo o modelo de soluções de Orquestração, Automação e Resposta de Segurança (SOAR) para lidar com ambientes heterogêneos.

[Applebaum et al. 2016] discutem CALDERA, um sistema automatizado de emulação de *red teams* focado nas ações de pós-comprometimento. O objetivo é superar as limitações de custo, repetibilidade e necessidade de expertise dos *red teams*, que avaliam a segurança imitando técnicas, táticas e procedimentos de adversários reais.

[de la Vallée et al. 2022] apresentam Sly, uma ferramenta de orquestração para automação de ataques cibernéticos em exercícios de *red teaming* que usa máquinas virtuais atacantes cujas ações são pré-definidas em grafos direcionados acíclicos.

Recentemente tem-se visto a combinação de arquiteturas e tecnologias baseadas em aprendizado de máquina e inteligência artificial para melhorar a segurança cibernética [Sarker 2021], onde a orquestração de segurança mostrou-se eficaz na detecção e prevenção de ataques, enfatizando a relevância de sistemas híbridos que combinam técnicas baseadas em assinaturas e anomalias para uma defesa mais robusta.

Outras ferramentas focam na análise de *malware* puramente automatizada, sem a possibilidade de criação de cenários em casos onde o *malware* a ser analisado precise de componentes internos (outros aplicativos ou bibliotecas) ou externos (*Websites* para obtenção de artefatos ou envio de dados) [Souza and Silva 2021, Botacin et al. 2021].

A solução mais próxima a este trabalho é a Cobalt Strike [HelpSystems 2024], uma ferramenta de simulação de adversários e operações de *red team* para realizar ataques simulados realistas e complexos em redes corporativas. Seu foco é em ataques manuais com suporte limitado a automação e dependência de interação humana. Além disso e da necessidade de alta *expertise*, Cobalt Strike é uma ferramenta comercial de alto custo.

3. Projeto e Implementação

O AMPOLA foi projetado como uma solução Web cujo intuito é permitir a geração de ambientes customizáveis e cenários de infeção realistas de maneira simples e eficiente. O código e a documentação do AMPOLA estão disponíveis em `https://github.com/ludersGabriel/tcc`. Diferente de outras ferramentas de automatização de segurança, AMPOLA foca em fornecer ao usuário total controle daquilo que está sendo analisado, cuidando apenas da criação e gerenciamento do ambiente com sistemas heterogêneos, deixando a análise dos artefatos ser pilotada pelo analista.

Dado o caráter customizável, a solução pode ser usada para uma variedade de cenários: desde análise direta de malware e simulação de fluxos de ataques até atividades *blue team/red team*. Além disso, AMPOLA permite a configuração de automatizações próprias, visto que gera máquinas virtuais de propósito geral a partir de ISOs fornecidas pelos operadores da solução. O projeto do AMPOLA consiste de três elementos básicos: um banco de dados, um *backend* e um *frontend*. O **banco de dados** é responsável por armazenar as informações relevantes à aplicação como dados dos usuários, informações relacionadas às máquinas virtuais cadastradas e situação das requisições. O banco foi implementado em PostgreSQL e é composto por quatro tabelas (Figura 1): usuários, máquinas virtuais, *requests* assíncronos e configurações de controle do sistema.

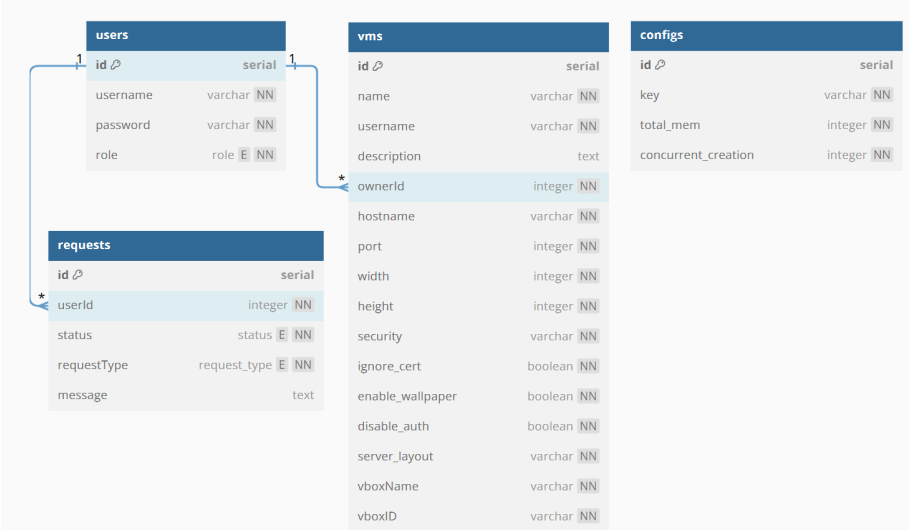

Figura 1. Esquema do banco de dados do AMPOLA

O ***backend*** é responsável por fornecer uma API capaz de expor e manipular os dados necessários do banco, além de fornecer uma maneira de controlar as máquinas virtuais a partir de um *websocket*, realizar autenticação de usuário e manter consistência das informações recebidas e enviadas. O principal objetivo do *backend* do AMPOLA é fornecer uma linha de conexão com as máquinas virtuais criadas no servidor, bem como impedir que usuários modifiquem informações que não são da sua autoria por meio de um *token* de autenticação e autorização. Para manipulação e recuperação dos dados do banco, o *backend* possui uma API Rest implementada em NodeJS (node) utilizando *express* para o servidor Web, além de utilizar DrizzleORM (`https://orm.drizzle.team/`) para interagir diretamente com o banco com segurança e consistência. Para fornecer acesso às máquinas virtuais, o *backend* expõe um servidor *websocket* responsável por manter um túnel entre o *frontend* e o *daemon* do *desktop gateway* Apache Guacamole (`https://guacamole.apache.org/`). Por sua vez, o Guacamole é quem faz a conexão direta às máquinas via o protocolo RDP. No contexto de orquestração de máquinas virtuais, foram implementados *scripts* em shell que utilizam `VBoxManage` (`https://www.virtualbox.org/manual/ch08.html`) para manipular as virtuais a partir do VirtualBox pela linha de comando. Vale ressaltar que as máquinas virtuais são criadas a partir de OVAS pré-montados—fornecidos pelo operador da aplicação—que, devido às peculiaridades do `VBoxManage`, precisam das *Guest Additions* e de um usuário administrador único sem senha para facilitar a interação.

O ***frontend*** é responsável por fornecer um painel de controle ao usuário com o intuito de criar, remover e gerenciar (desligar e ligar, por exemplo) as máquinas virtuais, assim como permitir acesso e controle direto dessas máquinas virtuais e fazer upload/download de arquivos. O *frontend* foi implementado utilizando React com bootstrap via Vite (`https://vitejs.dev/`), que é uma *Single Page Application* (SPA) responsável por permitir ao usuário interação e manipulação da aplicação. Cabe destacar a utilização da biblioteca `guacamole-common-js` (`https://guacamole.apache.org/doc/gug/guacamole-common-js.html`) para criar um túnel de conexão com o *daemon* presente no *backend* e permitir acesso às máquinas geradas a partir do navegador.

4. Estudos de Caso

A validação da AMPOLA foi feita considerando-se dois cenários, detalhados a seguir.

4.1. Cenário 1 - Usuário realiza *download* de *malware* e o executa

Para o primeiro caso, esquematiza-se um cenário simples, porém comum, de infecção devido à execução de software malicioso na vítima via *phishing* (Figura 2).

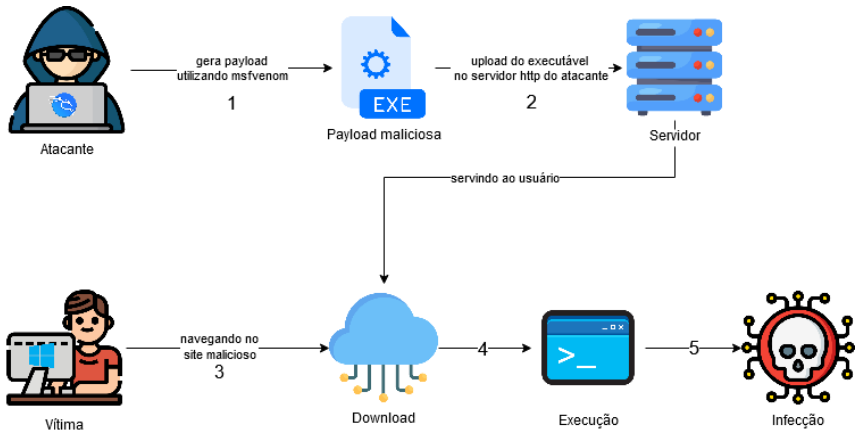

Figura 2. Esquema de infecção com duas VMs: Atacante/Servidor e Vítima

A Figura 3 mostra a máquina atacante servindo o executável e aguardando o *exploit*, enquanto que a Figura 4 mostra a visão da vítima. Nota-se que a conexão TCP foi um sucesso e que a máquina Atacante, além de listar o conteúdo de um diretório, também criou uma pasta chamada "`dir-from-kali`" no sistema da vítima, comprovando a infecção. É importante notar que *exploits* tão simples como o deste cenário são facilmente reconhecidos por soluções como o Windows Defender (desligado na máquina Vítima), corroborando a capacidade de customização da solução para estudo de diferentes ataques. Cenários como esse mostram a capacidade da plataforma de permitir simulações de ataques em ambientes vulneráveis, bem como estudar seu processo e impacto no alvo.

4.2. Cenário 2 - Ataques com *malware* multicomponente

Este cenário apresenta um exemplo de execução de ataques mais complexos, como os que causam infecções a partir de múltiplos componentes. Muitas vezes, as partes que compõem a infecção são capazes de evadir ferramentas de análise, impedindo concluir todo o fluxo e, consequentemente, dificultando a análise do ataque.

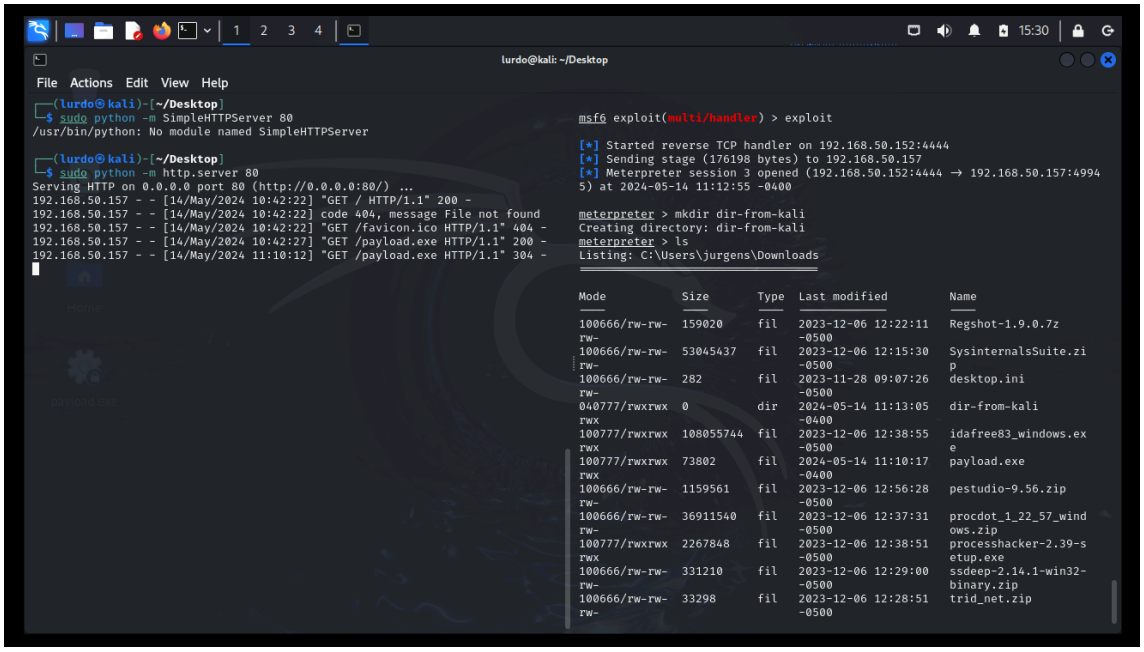

Figura 3. Visão da máquina Atacante

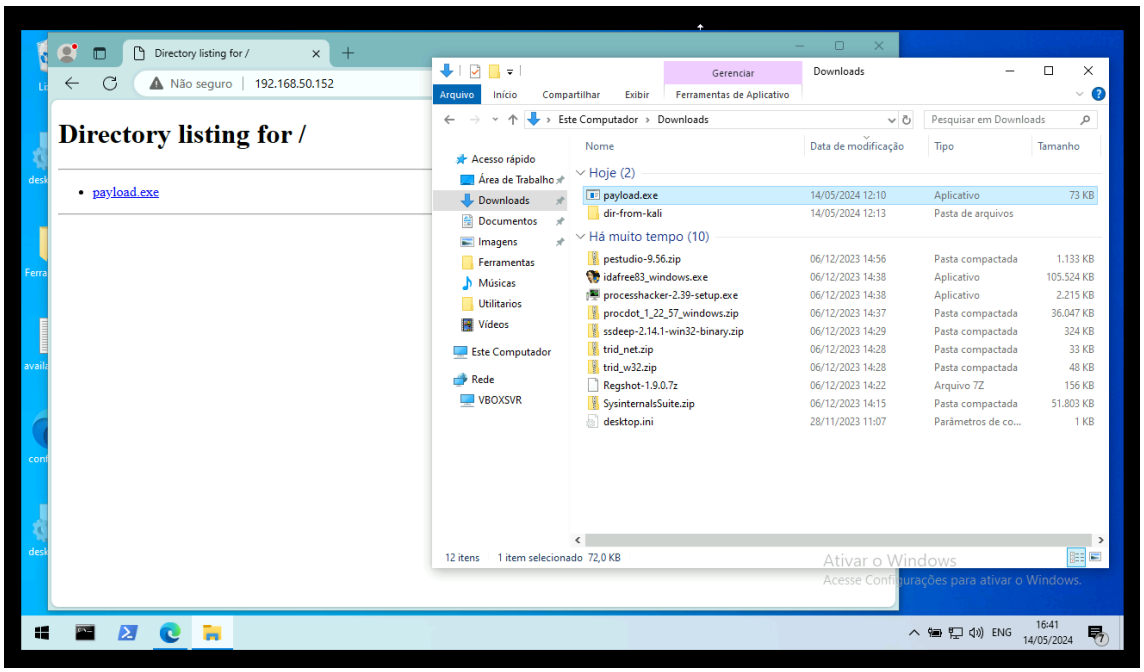

Figura 4. Visão da máquina Vítima

A Figura 5 apresenta o fluxo de um ataque simulado baseado em Remote Code Execution (RCE) utilizando o aplicativo de leitura de PDF FoxIt Reader (v2024.2.1.25153) [The Hacker News 2024]. Arquivos PDF maliciosos são de difícil identificação, já que tags como `/Launch` podem ser utilizadas de maneira benigna, e de fácil transmissão, uma vez que esses arquivos são aceitos sem problemas por praticamente todas as ferramentas de troca de mensagens e redes sociais existentes, podendo facilitar ataques por *phishing*.

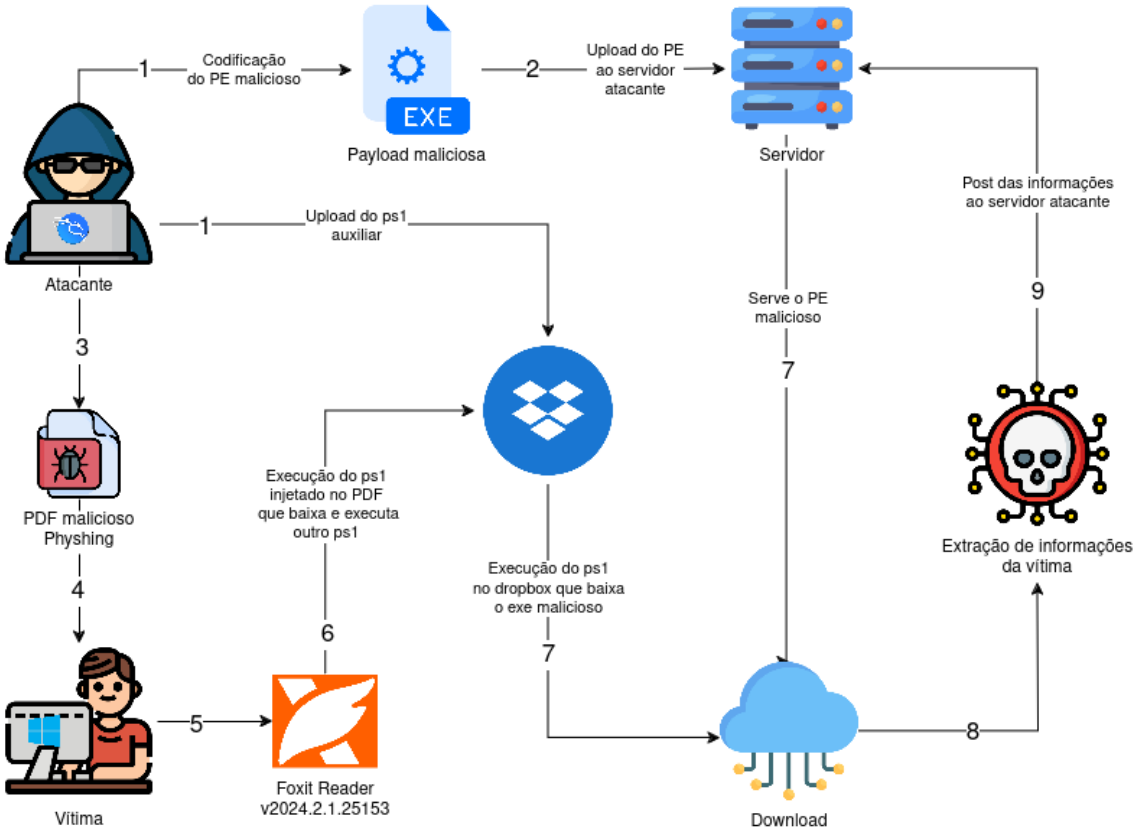

Figura 5. Fluxo de infecção com múltiplos componentes

Com AMPOLA, pode-se facilmente recriar um ambiente capaz de permitir a replicação de um ataque como esse, possibilitando o seu estudo de maneira mais profunda. Para tanto, criou-se uma VM Windows 10 com a versão desejada do *FoxIt*, uma VM Kali para ser um servidor HTTP simples e os seguintes arquivos:

- Um arquivo PE capaz de executar `ipconfig` e enviar as informações coletadas a um servidor qualquer na Internet.
- Um arquivo PowerShell auxiliar capaz de baixar e executar o PE malicioso do servidor atacante.
- Um PDF malicioso com código PowerShell capaz de baixar e executar o arquivo `.ps1` auxiliar de um servidor qualquer.

Com o `.ps1` auxiliar no *Dropbox*, o PE servido a partir da Kali e o PDF em mãos do usuário Windows, o fluxo dentro do AMPOLA é iniciado. Ao abrir o PDF com o FoxIt na máquina Vítima, obtém-se janelas de *pop-up* com opção positiva padrão (OK e Open, respectivamente ilustrados nas Figuras 6 e 7), propensas a serem aceitas sem leitura cuidadosa por usuários leigos. Ao clicar nos botões, o código PowerShell injetado na tag `/Launch` do pdf é executado, baixando e executando o `.ps1` do Dropbox. Esse `.ps1`, por sua vez, baixa o PE da máquina Kali (servidor atacante) e o executa. O PE extrai informações da máquina Vítima e as envia ao atacante (Figura 8), concluindo o ataque. Após a confirmação dos *pop-ups*, todo o resto da infecção é transparente ao usuário (exceto pelo piscar de um *prompt*). Com ofuscação, um atacante pode ser capaz de evadir defesas e realizar ações arbitrárias na máquina Vítima.

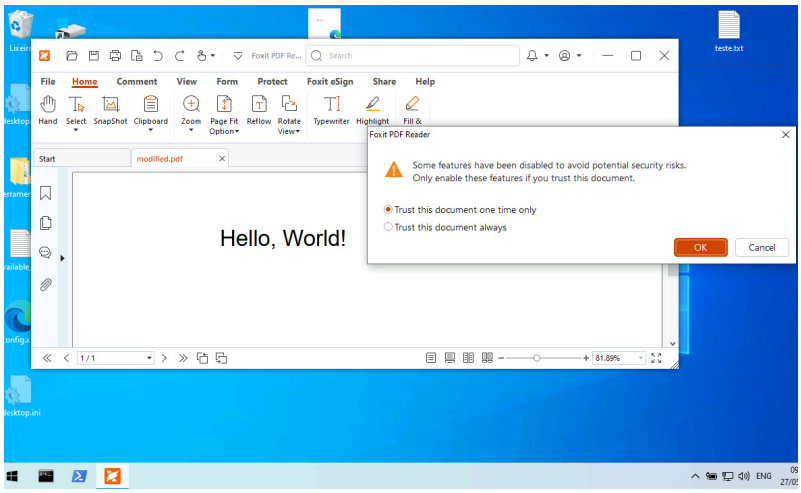

Figura 6. Pop-up do FoxIt com botão de OK marcado para confiar no documento.

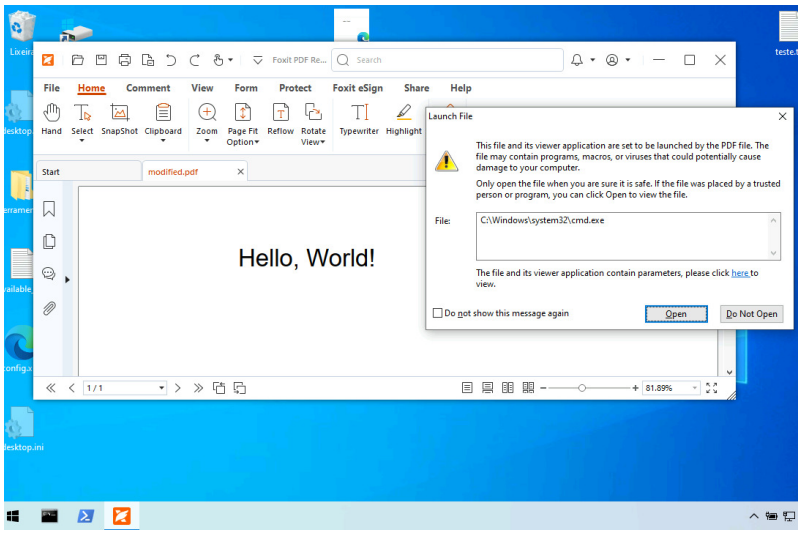

Figura 7. Pop-up do FoxIt com botão de Open marcado para executar arquivo.

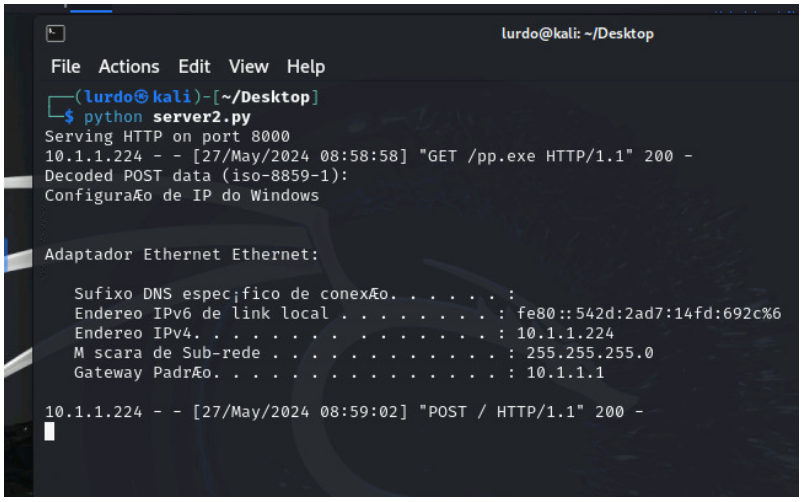

Figura 8. Servidor Web atacante recebendo informações da vítima (ipconfig).

5. Conclusão

Com a populariade de malwares complexos e ataques cada vez mais criativos, é clara a necessidade de um sistema de orquestração capaz de fornecer a possibilidade de criação de ambientes controlados de maneira rápida e simplificada. Para tanto, este artigo introduziu AMPOLA, uma ferramenta com interface Web e foco em criação e gerenciamento de máquinas virtuais para geração de cenários de ataque por *malware*. Além disso, descreveu-se dois cenários de ataques cada vez mais comuns e presentes no cotidiano cibernético, com explicação dos ambientes criados dentro do AMPOLA.

Referências

Applebaum, A., Miller, D., Strom, B., Korban, C., and Wolf, R. (2016). Intelligent, automated red team emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ACSAC '16, page 363–373, New York, NY, USA. Association for Computing Machinery.

Botacin, M., Ceschin, F., and Grégio, A. (2021). Corvus: Uma solução sandbox e de threat intelligence para identificação e análise de malware. In *Anais Estendidos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 50–57, Porto Alegre, RS, Brasil. SBC.

de la Vallée, P., Iosifidis, G., and Mees, W. (2022). Cyber red teaming: Overview of sly, an orchestration tool. *Information Security: An International Journal*, 53(2):273–286.

Gatlan, S. (2024). Microsoft fixes windows zero-day exploited in qakbot malware attacks. `https://www.bleepingcomputer.com/news/security/microsoft-fixes-windows-zero-day-exploited-in-qakbot-malware-attacks/`. Accessado: 06/2024.

HelpSystems (2024). Cobalt strike: Adversary simulation and red team operations. `https://www.cobaltstrike.com/`. Accessado: 06/2024.

Lee, M., Jang-Jaccard, J., and Kwak, J. (2022). Novel architecture of security orchestration, automation and response in internet of blended environment. *Computers, Materials & Continua*, 73(1):199–223.

Ray, H., Vemuri, R., and Kantubhukta, H. (2005). Toward an automated attack model for red teams. *IEEE Security Privacy*, 3(4):18–25.

Sarker, I. H. (2021). Ai-driven cybersecurity: An overview, security intelligence modeling and research directions. *Arxiv Preprints*.

Souza, C. and Silva, F. (2021). Freki: Uma ferramenta para análise automatizada de malware. In *Anais Estendidos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 58–65, Porto Alegre, RS, Brasil. SBC.

The Hacker News (2024). Foxit pdf reader flaw exploited by attackers. `https://thehackernews.com/2024/05/foxit-pdf-reader-flaw-exploited-by.html`. Accessado: 05/2024.